

AMENDMENTS TO THE CLAIMS

Please amend claim 1-29, as shown below.

1 1. (Currently amended) A method for emulating a plurality of virtual timers in a
2 virtual computer system operating on a physical computer, the physical computer
3 having one or more timers for keeping track of a real time for the physical computer, the
4 virtual timers being programmable by guest software to generate a plurality of timer
5 events, the method comprising:

6 receiving programming information from the guest software for
7 programming a first virtual timer;

8 receiving programming information from the guest software for
9 programming a second virtual timer;

10 determining when the first virtual timer is set to would generate timer
11 events according to the real time if it were implemented in a physical computer
12 system, based on the programming information received from the guest software;

13 determining when the second virtual timer is set to would generate timer
14 events according to the real time if it were implemented in a physical computer
15 system, based on the programming information received from the guest software;
16 and

17 wherein the generation of timer events falls behind the real time, so that a
18 first plurality of timer events, including one or more timer events of the first virtual
19 timer and one or more timer events of the second virtual timer, are set to have
20 already occurred according to the real time, but the first plurality of timer events
21 have not yet occurred in the virtual computer system; and

22 generating the first plurality of timer events for the first virtual timer and the
23 second virtual timer in the same combined sequence as the timer events are set
24 to occur according to the real time that they would occur if the first and second
25 virtual timers were implemented in a physical computer system.

1 2. (Currently amended) ~~The method of claim 1;~~ A method for emulating a
2 plurality of virtual timers in a virtual computer system operating on a physical computer,
3 the physical computer having one or more timers for keeping track of a real time for the

4 physical computer, the virtual timers being programmable by guest software to generate
5 a plurality of timer events, the method comprising:

6 receiving programming information from the guest software for
7 programming a first virtual timer;

8 receiving programming information from the guest software for
9 programming a second virtual timer;

10 determining when the first virtual timer is set to generate timer events
11 according to the real time, based on the programming information received from
12 the guest software;

13 determining when the second virtual timer is set to generate timer events
14 according to the real time, based on the programming information received from
15 the guest software;

16 wherein the generation of timer events falls behind the real time, so that a
17 first plurality of timer events, including one or more timer events of the first virtual
18 timer and one or more timer events of the second virtual timer, are set to have
19 already occurred according to the real time, but the first plurality of timer events
20 have not yet occurred in the virtual computer system; and

21 generating the first plurality of timer events in the same combined
22 sequence as the timer events are set to occur according to the real time,

23 wherein a catch-up mode is used when the generation of timer events in
24 the virtual computer system is behind the real time, timing-of-when-the-timer
25 events-would-be-generated-in-a-physical-computer-system-and

26 wherein a normal mode is used when the generation of timer events in the
27 virtual computer system is caught up to the real time, timing-of-when-the-timer
28 events-would-be-generated-in-a-physical-computer-system;

29 wherein, when the catch-up mode is used, the average rate of timer
30 events in the virtual computer system exceeds the average rate at which timer
31 events are set to occur according to the real time, would-be-generated-in-a
32 physical-computer-system; and

wherein, when the normal mode is used, the average rate of timer events in the virtual computer system is substantially the same as the average rate at which timer events are set to occur according to the real time would be generated in a physical computer system.

3. (Currently amended) The method of claim 1; A method for emulating a plurality of virtual timers in a virtual computer system operating on a physical computer, the physical computer having one or more timers for keeping track of a real time for the physical computer, the virtual timers being programmable by guest software to generate a plurality of timer events, the method comprising:

receiving programming information from the guest software for programming a first virtual timer;

receiving programming information from the guest software for programming a second virtual timer;

determining when the first virtual timer is set to generate timer events according to the real time, based on the programming information received from the guest software;

determining when the second virtual timer is set to generate timer events according to the real time, based on the programming information received from the guest software; and

generating timer events for the first virtual timer and the second virtual timer in the same combined sequence as the timer events are set to occur according to the real time.

wherein a catch-up mode is used when the generation of timer events in the virtual computer system is behind the real time, timing of when the timer events would be generated in a physical computer system and

wherein a normal mode is used when the generation of timer events in the virtual computer system is caught up to the real time, timing of when the timer events would be generated in a physical computer system;

25 wherein, when the catch-up mode is used, the average rate of timer
26 events in the virtual computer system exceeds the average rate at which timer
27 events are set to occur according to the real time, ~~would be generated in a~~
28 ~~physical computer system~~; and

29 wherein, when the normal mode is used, the average rate of timer events
30 in the virtual computer system is substantially the same as the average rate at
31 which timer events are set to occur according to the real time ~~would be~~
32 ~~generated in a physical computer system~~.

1 4. (Currently amended) The method of claim 3 ~~[[2]]~~, wherein, when the
2 normal mode is used, the interval between successive timer events in the virtual
3 computer system is substantially the same as the interval ~~that would occur~~ between the
4 same successive timer events as set according to the real time in a physical computer
5 ~~system~~.

1 5. (Currently amended) The method of claim 3 ~~[[2]]~~, wherein the catch-up
2 mode is entered when the generation of timer events in the virtual computer system falls
3 behind the real time ~~timing of when the timer events would be generated in a physical~~
4 ~~computer system~~ by a predetermined amount and the normal mode is entered when the
5 generation of timer events in the virtual computer system goes ahead of the real time
6 ~~timing of when the timer events would be generated in a physical computer system~~ by a
7 predetermined amount.

1 6. (Currently amended) The method of claim 3 ~~[[2]]~~, wherein the catch-up
2 mode is entered substantially immediately when the generation of timer events in the
3 virtual computer system falls behind the real time ~~timing of when the timer events would~~
4 ~~be generated in a physical computer system~~ and the normal mode is entered
5 substantially immediately when the generation of timer events in the virtual computer
6 system catches up to the real time ~~timing of when the timer events would be generated~~
7 ~~in a physical computer system~~.

1 7. (Currently amended) The method of claim 3 [[1]], wherein, if the guest
2 software attempts to read a read count value from a virtual timer, a returned count value
3 is returned to the guest software that represents a returned time value that occurs after
4 a preceding time value that is represented by a most recent preceding timer event and
5 before a next time value that is represented by a next timer event to occur.

1 8. (Currently amended) The method of claim 7, wherein the returned time
2 value ~~that is represented by the count value that is returned to the guest software falls~~
3 proportionately between the preceding time value ~~that is represented by the most recent~~
4 ~~preceding timer event and the next time value that is represented by the next timer~~
5 ~~event to occur, based on the proportion at which the real time of the attempted reading~~
6 ~~of the read count value falls between the real actual time that at which the most recent~~
7 ~~preceding timer event was generated and the real actual time that at which the next~~
8 timer event is scheduled to be generated.

1 9. (Currently amended) The method of claim 3 [[1]], wherein the method is
2 performed by keeping track of an apparent time, which represents the time as it would
3 appear to the guest software, as indicated by the virtual timers.

1 10. (Currently amended) The method of claim 3 [[1]], wherein the method is
2 performed using a timer event queue.

1 11. (Currently amended) A computer program embodied in a tangible
2 computer-readable storage medium, the computer program being executable on a
3 physical computer as part of a virtual computer system, the physical computer having
4 one or more timers for keeping track of a real time for the physical computer, the virtual
5 computer system comprising one or more timer emulators for emulating a plurality of
6 virtual timers, each of the plurality of virtual timers generating one or more timer events,
7 the computer program comprising:

8 a time coordinator for coordinating the respective timer events of the
9 plurality of virtual timers, the time coordinator:

10 determining how each of the plurality of virtual timers has been
11 programmed;

12 based on how each of the virtual timers has been programmed,
13 determining a relative sequence of timer events as set according to the
14 real time, that would be generated by the virtual timers if they were
15 implemented in a physical computer system; and

16 wherein the generation of timer events falls behind the real time, so
17 that a first plurality of timer events, including a timer event from each of at
18 least two of the virtual timers, are set to have already occurred according
19 to the real time, but the first plurality of timer events have not yet occurred
20 in the virtual computer system; and

21 notifying the one or more timer emulators when each of the plurality
22 of virtual timers is to should generate a timer event, so that the first
23 plurality of timer events are generated in the same combined sequence as
24 the timer events are set to occur according to the real time if the virtual
25 timers had been implemented in a physical computer system.

1 12. (Currently amended) The computer program of claim 11, wherein the time
2 coordinator uses an apparent time, representing the time as it appears to a software
3 entity that is using the plurality of virtual timers, in determining the relative sequence of
4 timer events according to the real time that would be generated by the virtual timers if
5 they were implemented in a physical computer system.

1 13. (Currently amended) The computer program of claim 11, wherein the time
2 coordinator uses a timer event queue to determine the relative sequence of timer events
3 according to the real time that would be generated by the virtual timers if they were
4 implemented in a physical computer system.

1 14. (Currently amended) The computer program of claim 13, wherein the
2 timer event queue maintains a single time value for each of the plurality of virtual timers,
3 representing a time at which the respective virtual timer is to should generate its next
4 timer event.

1 15. (Previously presented) The computer program of claim 14, wherein the
2 timer event queue comprises a linked list.

1 16. (Currently amended) ~~The computer program of claim 14;~~ A computer
2 program embodied in a computer-readable storage medium, the computer program
3 being executable on a physical computer as part of a virtual computer system, the
4 physical computer having one or more timers for keeping track of a real time for the
5 physical computer, the virtual computer system comprising one or more timer emulators
6 for emulating a plurality of virtual timers, each of the plurality of virtual timers generating
7 one or more timer events, the computer program comprising:

8 a time coordinator for coordinating the respective timer events of the
9 plurality of virtual timers, the time coordinator:

10 determining how each of the plurality of virtual timers has been
11 programmed;

12 based on how each of the virtual timers has been programmed,
13 determining a relative sequence of timer events as set according to the
14 real time; and

15 notifying the one or more timer emulators when each of the plurality
16 of virtual timers is to generate a timer event, so that the timer events are
17 generated in the same combined sequence as the timer events are set to
18 occur according to the real time,

19 wherein the time coordinator has a catch-up mode that is used when the
20 generation of timer events in the virtual computer system is behind the real time,
21 ~~timing of when the timer events would be generated in a physical computer~~
22 ~~system and a normal mode that is used when the generation of timer events in~~

the virtual computer system is caught up to the real time timing-of-when the timer events would be generated in a physical computer system;

wherein, when the time coordinator is in the catch-up mode, the average rate of timer events in the virtual computer system exceeds the average rate at which timer events are set to occur according to the real time, would be generated in a physical computer system; and

wherein, when the time coordinator is in the normal mode, the average rate of timer events in the virtual computer system is substantially the same as the average rate at which timer events are set to occur according to the real time would be generated in a physical computer system.

17. (Currently amended) The computer program of claim 16, wherein, when the time coordinator is in the catch-up mode, the interval between successive timer events in the virtual computer system is substantially proportional to the interval that would occur between the same successive timer events as set according to the real time in a physical computer system.

18. (Currently amended) The computer program of claim 16, wherein, when the time coordinator is in the normal mode, the interval between successive timer events in the virtual computer system is substantially the same as the interval that would occur between the same successive timer events as set according to the real time in a physical computer system.

19. (Currently amended) The computer program of claim 16, wherein the time coordinator enters the catch-up mode when the generation of timer events in the virtual computer system falls behind the real time timing-of-when the timer events would be generated in a physical computer system by a predetermined amount and the time coordinator enters the normal mode when the generation of timer events in the virtual computer system goes ahead of the real time timing-of-when the timer events would be generated in a physical computer system by a predetermined amount.

1 20. (Currently amended) The computer program of claim 16, wherein the time
2 coordinator enters the catch-up mode substantially immediately when the generation of
3 timer events in the virtual computer system falls behind the real time ~~timing of when the~~
4 ~~timer events would be generated in a physical computer system~~ and the time
5 coordinator enters the normal mode substantially immediately when the generation of
6 timer events in the virtual computer system catches up to the real time ~~timing of when~~
7 ~~the timer events would be generated in a physical computer system.~~

1 21. (Currently amended) The computer program of claim 16 ~~[[11]]~~, wherein, if
2 a software entity attempts to read a read count value from a virtual timer, the time
3 coordinator provides a value to one of the timer emulators, which causes the timer
4 emulator to return a returned count value to the software entity that represents a
5 returned time value that occurs after a preceding time value that is represented by a
6 most recent preceding timer event and before a next time value that is represented by a
7 next timer event to occur.

1 22. (Currently amended) The computer program of claim 21, wherein the
2 ~~returned time value that is represented by the count value that is returned to the~~
3 ~~software entity falls proportionately between the~~ preceding time value that is
4 ~~represented by the most recent preceding timer event and the~~ next time value that is
5 ~~represented by the next timer event to occur, based on the proportion at which the~~ real
6 time of the attempted reading of the count value falls between the real actual time that
7 ~~at which~~ the most recent preceding timer event was generated and the real actual time
8 ~~that at which~~ the next timer event is scheduled to be generated.

1 23. (Currently amended) A method for coordinating a plurality of virtual timers
2 in a virtual computer system, the virtual computer system operating within a physical
3 computer system, the physical computer system having one or more timers for keeping
4 track of a real time for the physical computer system, the method comprising:

5 receiving programming information for each of the virtual timers, indicating
6 when each of the virtual timers is to generate timer events;
7 determining when each of the virtual timers would is set to generate timer
8 events according to the real time, if the virtual timers were implemented in a
9 physical computer system; and
10 wherein the virtual timers fall behind the real time, so that a first plurality of
11 timer events, including a timer event from each of at least two of the virtual
12 timers, are set to have already occurred according to the real time, but the first
13 plurality of timer events have not yet occurred in the virtual computer system; and
14 causing the virtual timers to generate the first plurality of timer events in
15 the same combined sequence as the timer events are set to occur according to
16 the real time if the virtual timers had been implemented in a physical computer
17 system.

1 24. (Currently amended) ~~The method of claim 23, further comprising: A~~
2 method for coordinating a plurality of virtual timers in a virtual computer system, the
3 virtual computer system operating within a physical computer system, the physical
4 computer system having one or more timers for keeping track of a real time for the
5 physical computer system, the method comprising:

6 receiving programming information for each of the virtual timers, indicating
7 when each of the virtual timers is to generate timer events;

8 determining when each of the virtual timers is set to generate timer events
9 according to the real time;

10 causing the virtual timers to generate timer events in the same combined
11 sequence as the timer events are set to occur according to the real time;

12 using a physical timer in the physical computer system to determine a real
13 time reference that progresses in accordance with the timing of the physical
14 timer;

15 determining an apparent time that would appears to exist within the virtual
16 computer system based on timing information provided by the virtual timers;

when the apparent time is substantially the same as the real time, generating timer events at substantially the same real time as the timer events are set to occur according to the real time they would be generated if the virtual timers had been implemented in a physical computer system; and

when the apparent time is substantially behind the real time, generating timer events at a faster rate than the timer events are set to occur according to the real time they would be generated if the virtual timers had been implemented in a physical computer system, until the apparent time catches up to the real time.

25. (Currently amended) The method of claim 24, wherein, when the apparent time is substantially behind the real time, the interval between successive timer events in the virtual computer system is substantially proportional to the interval that would occur between the same successive timer events as set according to the real time in a physical computer system.

26. (Currently amended) The method of claim 24, wherein timer events are generated at a faster rate than the timer events are set to occur according to the real time they would be generated if the virtual timers had been implemented in a physical computer system, when the apparent time falls behind the real time by a predetermined amount; and wherein timer events are generated at substantially the same real time as the timer events are set to occur according to the real time they would be generated if the virtual timers had been implemented in a physical computer system, when the apparent time goes ahead of the real time by a predetermined amount.

27. (Currently amended) The method of claim 24, wherein timer events are generated at a faster rate than the timer events are set to occur according to the real time they would be generated if the virtual timers had been implemented in a physical computer system substantially immediately when the apparent time falls behind the real time, and wherein timer events are generated at substantially the same real time as the

6 ~~timer events are set to occur according to the real time they would be generated if the~~
7 ~~virtual timers had been implemented in a physical computer system substantially~~
8 ~~immediately when the apparent time catches up to the real time.~~

1 28. (Currently amended) The method of claim 24 ~~[[23]]~~, wherein, if a software
2 entity within the virtual computer system attempts to read a read count value from a
3 virtual timer, a returned count value is returned to the software entity that represents a
4 returned time value that occurs after a preceding time value that is represented by a
5 most recent preceding timer event and before a next time value that is represented by a
6 next timer event to occur.

1 29. (Currently amended) The method of claim 28, wherein the returned time
2 value ~~that is represented by the count value that is returned to the software entity~~ falls
3 proportionately between the preceding time value ~~that is represented by the most recent~~
4 ~~preceding timer event~~ and the next time value ~~that is represented by the next timer~~
5 ~~event to occur~~, based on the proportion at which the real time of the attempted reading
6 of the read count value falls between the real actual time that at which the most recent
7 preceding timer event was generated and the real actual time that at which the next
8 timer event is scheduled to be generated.